ЛАБОРАТОРНАЯ РАБОТА №8

Математические основы защиты информации и информационной безопасности. Целочисленная арифметика многократной точности

Кузнецов Юрий Владимирович

# Введение

В данной работе рассмотрим алгоритмы для выполнения арифметических операций с большими целыми числами.

# Основное содержание

## Алгоритм:

* Алгоритм 1(сложение неотрицательных целых чисел).
* Алгоритм 2(вычитание неотрицательных целых чисел).
* Алгоритм 3 (умножение неотрицательных целых чисел столбиком).
* Алгоритм 4(быстрый столбик).

# Кодовая реализация

## Алгоритм 1(сложение неотрицательных целых чисел).

function add\_large\_numbers(a, b, base=10)  
 carry = 0  
 a\_digits = parse.(Int, collect(string(a)))  
 b\_digits = parse.(Int, collect(string(b)))  
 len\_a = length(a\_digits)  
 len\_b = length(b\_digits)  
 max\_len = max(len\_a, len\_b)  
 result = zeros(Int64, max\_len + 1)

if len\_a < len\_b  
 a\_digits = vcat(zeros(Int64, len\_b - len\_a), a\_digits)  
 elseif len\_b < len\_a  
 b\_digits = vcat(zeros(Int64, len\_a - len\_b), b\_digits)  
 end

for i in reverse(1:max\_len)  
 temp = (a\_digits[i] + b\_digits[i] + carry) % base  
 result[i+1] = temp  
 carry = (a\_digits[i] + b\_digits[i] + carry) ÷ base  
 end  
  
 result[1] = carry  
 return parse(Int, join(string.(result)))  
end  
  
println(add\_large\_numbers(87452, 1238))

## Алгоритм 2(вычитание неотрицательных целых чисел).

function subtract\_large\_numbers(a, b, base=10)  
 if a < b  
 return "$a is smaller than $b"  
 end  
  
 a\_digits = parse.(Int, collect(string(a)))  
 b\_digits = parse.(Int, collect(string(b)))  
 len\_a = length(a\_digits)  
 len\_b = length(b\_digits)  
 max\_len = max(len\_a, len\_b)  
 result = zeros(Int64, max\_len)  
  
 if len\_b < len\_a  
 b\_digits = vcat(zeros(Int64, len\_a - len\_b), b\_digits)  
 end

for i in reverse(1:max\_len)  
 if a\_digits[i] < b\_digits[i]  
 a\_digits[i-1] -= 1  
 a\_digits[i] += base  
 end  
 result[i] = a\_digits[i] - b\_digits[i]  
 end  
  
 return parse(Int, join(string.(result)))  
end  
  
println(subtract\_large\_numbers(87452, 1238))

## Алгоритм 3 (умножение неотрицательных целых чисел столбиком).

function multiply\_large\_numbers(a, b, base=10)  
 a\_digits = parse.(Int, collect(string(a)))  
 b\_digits = parse.(Int, collect(string(b)))  
 len\_a = length(a\_digits)  
 len\_b = length(b\_digits)  
 result = zeros(Int64, len\_a + len\_b)

for j in reverse(1:len\_b)  
 carry = 0  
 for i in reverse(1:len\_a)  
 temp = a\_digits[i] \* b\_digits[j] + result[i+j] + carry  
 result[i+j] = temp % base  
 carry = temp ÷ base  
 end  
 result[j] = carry  
 end  
  
 return parse(Int, join(string.(result)))  
end  
  
println(multiply\_large\_numbers(87452, 1238))

## Алгоритм 4(быстрый столбик).

function multiply\_fast(a, b, base=10)  
 a\_digits = parse.(Int, collect(string(a)))  
 b\_digits = parse.(Int, collect(string(b)))  
 len\_a = length(a\_digits)  
 len\_b = length(b\_digits)  
 result = zeros(Int64, len\_a + len\_b)  
 temp = 0

for s in 0:len\_a+len\_b-1  
 for i in 0:s  
 if len\_a-i <= 0 || len\_b-s+i <= 0  
 continue  
 end  
 temp += a\_digits[len\_a-i] \* b\_digits[len\_b-s+i]  
 end  
 result[len\_a+len\_b-s] = temp % base  
 temp = temp ÷ base  
 end  
  
 return parse(Int, join(string.(result)))  
end  
  
println(multiply\_fast(87452, 1238))

# Заключение

В данной лабораторной работе были реализованы алгоритмы для выполнения арифметических операций с большими целыми числами.